Profiling Production OpenSHMEM Applications

John C. Linford, Samuel Khuvis,
Sameer Shende, Allen Malony
ParaTools, Inc.

Neena Imam,
Manjunath Gorentla Venkata
ORNL

OpenSHMEM'16
2 August 2016
Baltimore, MD

ParaTools

Performance in Production

- Characterize the performance of production applications operating at **large scales without modifying** the application or relying on **debugging symbols** or interfaces like **PSHMEM**.
- Production ➔ reduced tool support.
  - PSHMEM and similar unlikely to be present.
  - New performance bugs likely to appear.

Copyright © ParaTools, Inc.

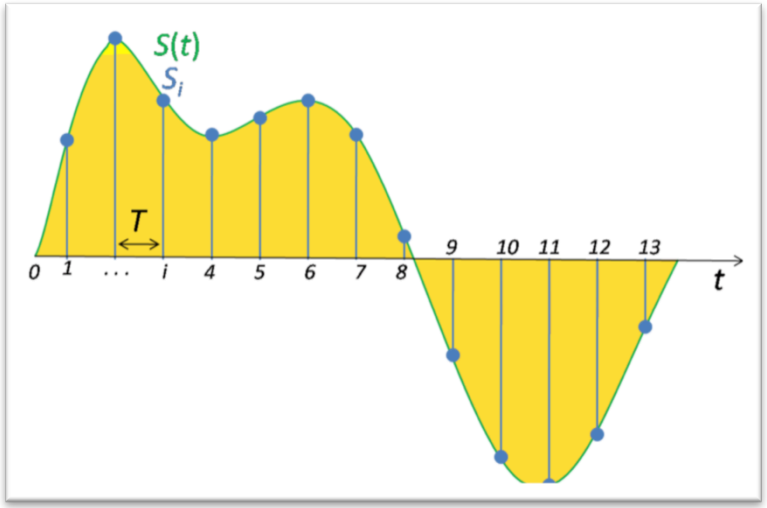
Performance Data Measurement

Direct via Probes

```
call TAU_START('potential')
// code
call TAU_STOP('potential')
```

- Exact measurement
- Fine-grain control
- Calls inserted into code

Indirect via Sampling



- No code modification
- Minimal effort
- Relies on debug symbols (**-g** option)

Copyright © ParaTools, Inc.

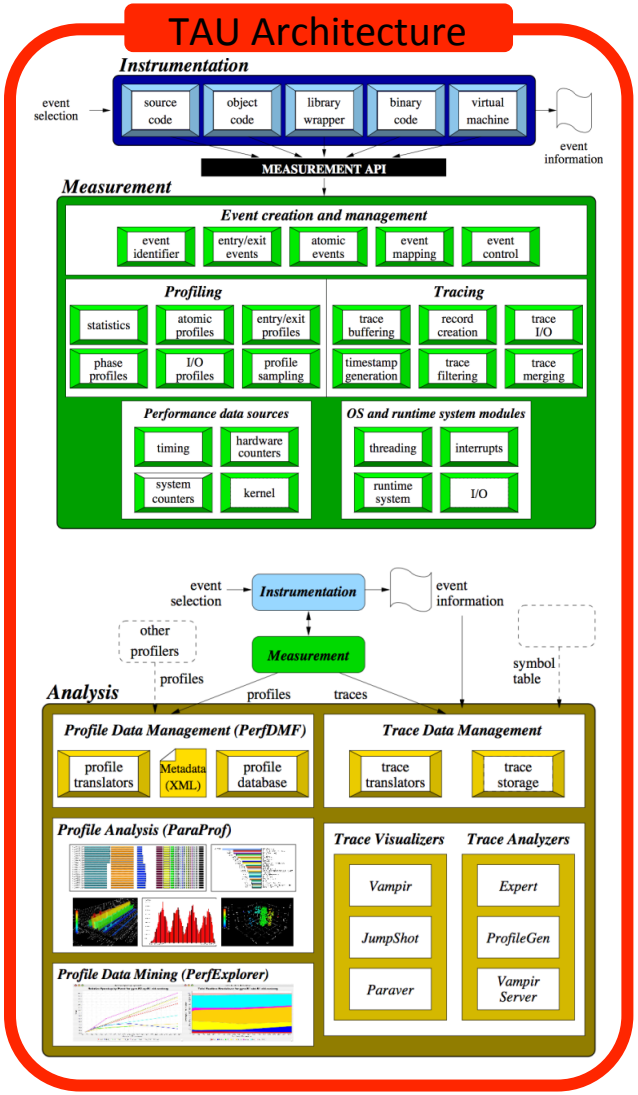
Approach: Generate a Wrapper Library

- Parse SHMEM headers and discover SHMEM API.
- Automatically generate a wrapper library.
  - Obviates PSHMEM
  - No debugging symbols required
- Use the linker to intercept SHMEM calls
  - Dynamic: LD_PRELOAD
  - Static: ld -wrap flag
- Build into the TAU Performance System®
  - Works well in production environments
  - Low runtime overhead (~4%)
  - Scalable

Copyright © ParaTools, Inc.

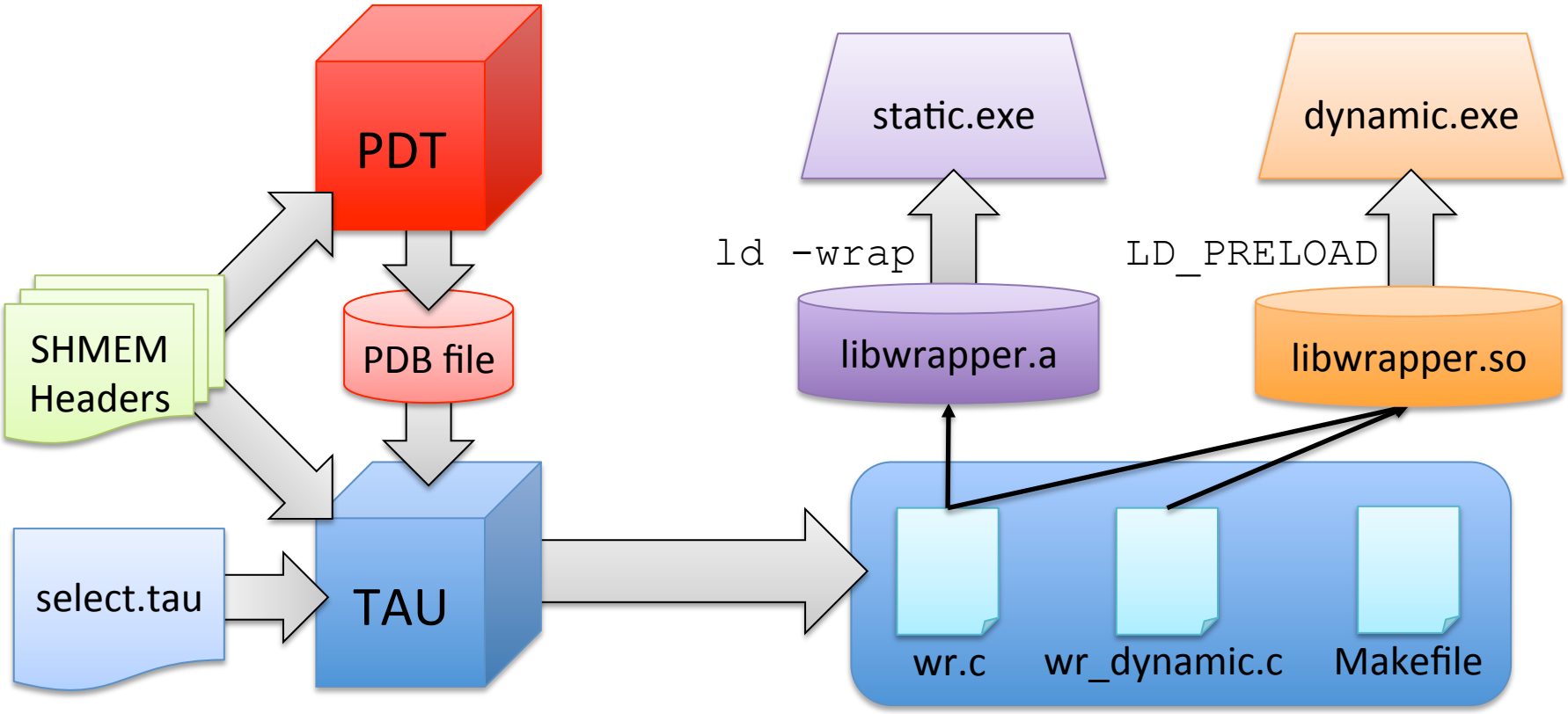
The TAU Performance System®

- ~25 year project
- *Integrated toolkit* for performance problem solving
  - Instrumentation, measurement, analysis, visualization
  - Portable profiling and tracing
  - Performance data management and data mining
- Direct and indirect measurement
- *Free, open source, BSD license*
- Available on all HPC platforms (and some non-HPC)
- http://tau.uoregon.edu/



Copyright © ParaTools, Inc.

SHMEM Wrapper Library Generation

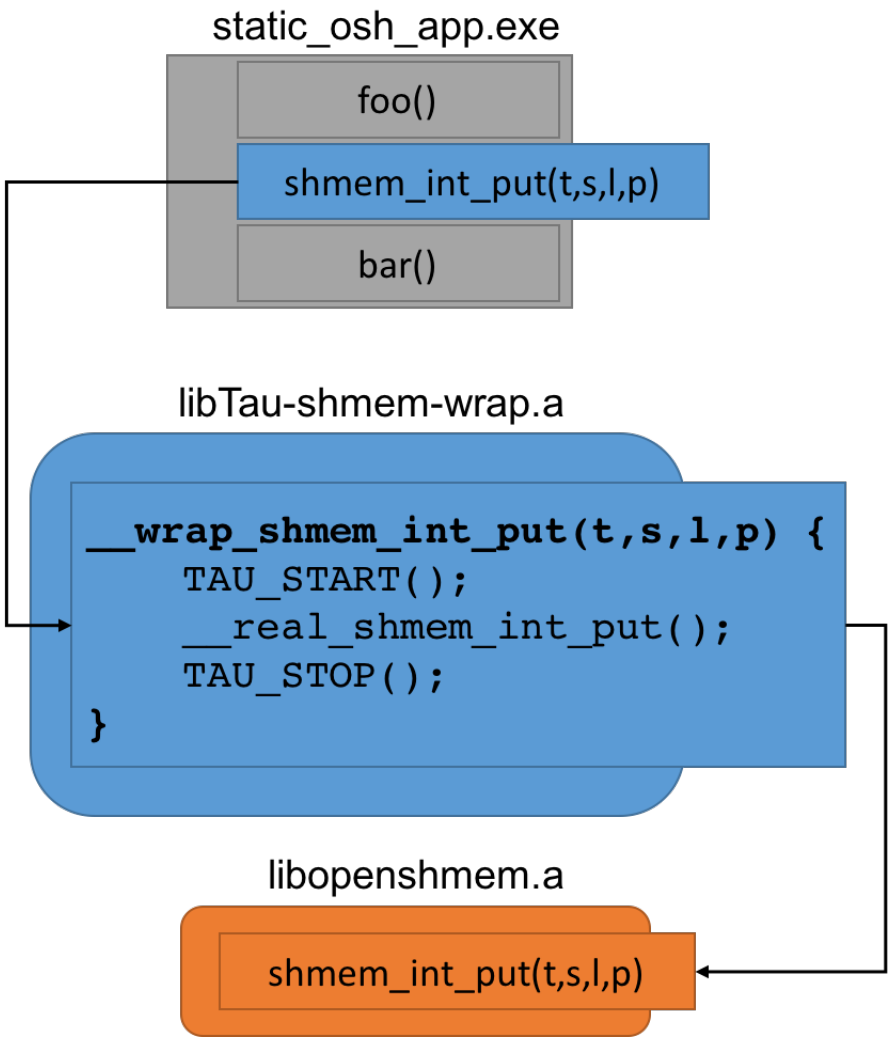


Copyright © ParaTools, Inc.

Symbol Wrapping

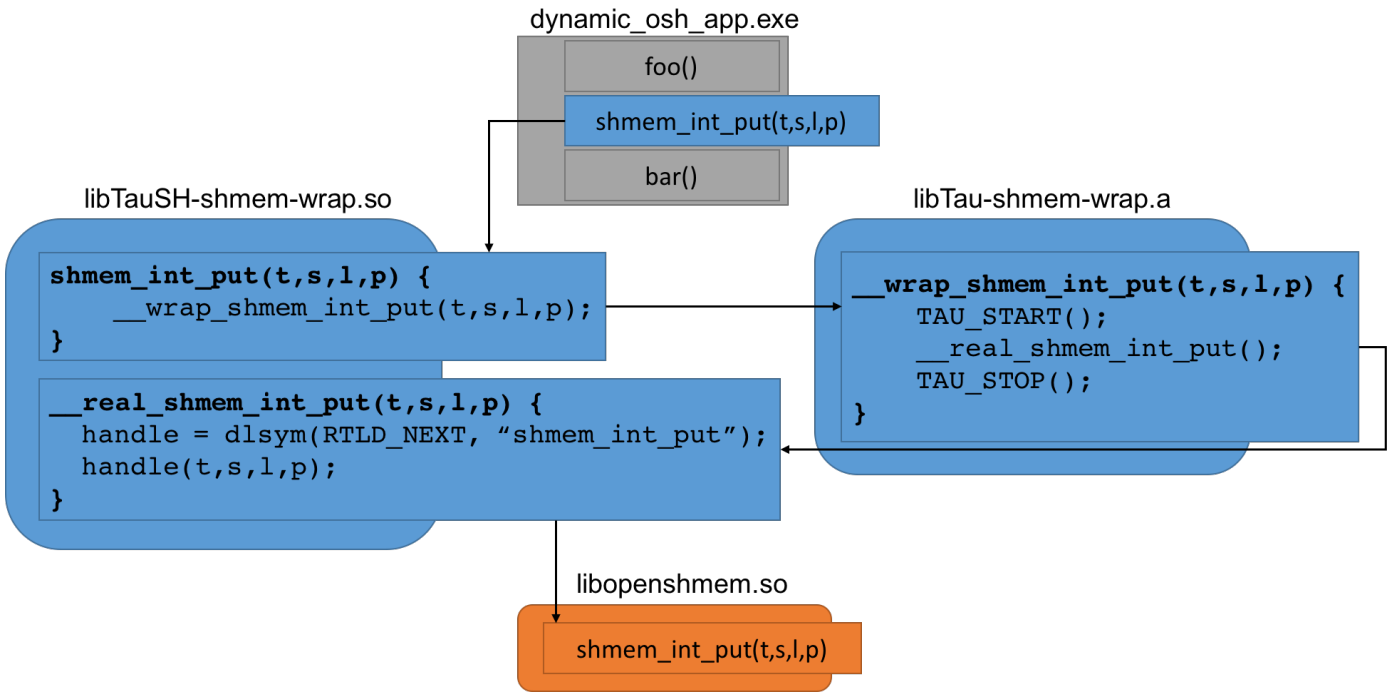
- Use the `-wrap` linker option
- Undefined reference to "`sym`" resolves to "`__wrap_sym`"
- Undefined reference to "`__real_sym`" resolves to "`sym`"
- Use TAU compiler wrapper scripts with -optLinkOnly, e.g.

```
tau_ftn.sh \
    -optLinkOnly \
    -o static.exe *.o
```



Copyright © ParaTools, Inc.

Library Preloading



- Unresolved reference to "`sym`" binds to "`sym`" in wrapper library, which calls "`__wrap_sym`" implemented in the static wrapper.
- Unresolved reference to "`__real_sym`" in static wrapper binds to "`__real_sym`" defined in dynamic wrapper.
- "`__real_sym`" uses dynamic linker API to locate "`sym`" and call it.
- Use tau_exec:
  - **`oshrun -np 256 tau_exec –T serial -shmem ./foo.exe`**

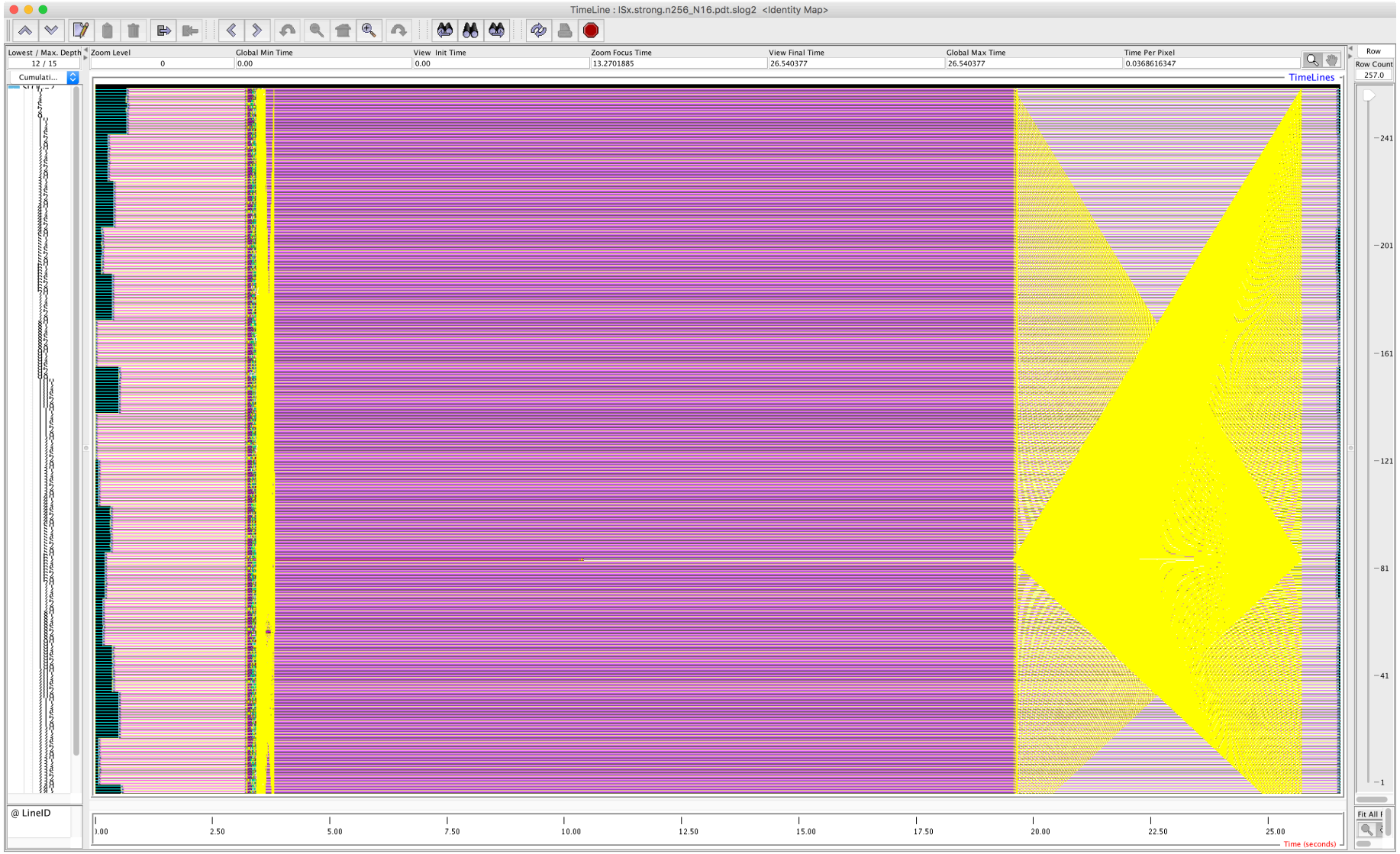
Copyright © ParaTools, Inc.

Profiling Production OpenSHMEM Applications

PERFORMANCE DATA ANALYSIS

Copyright © ParaTools, Inc.

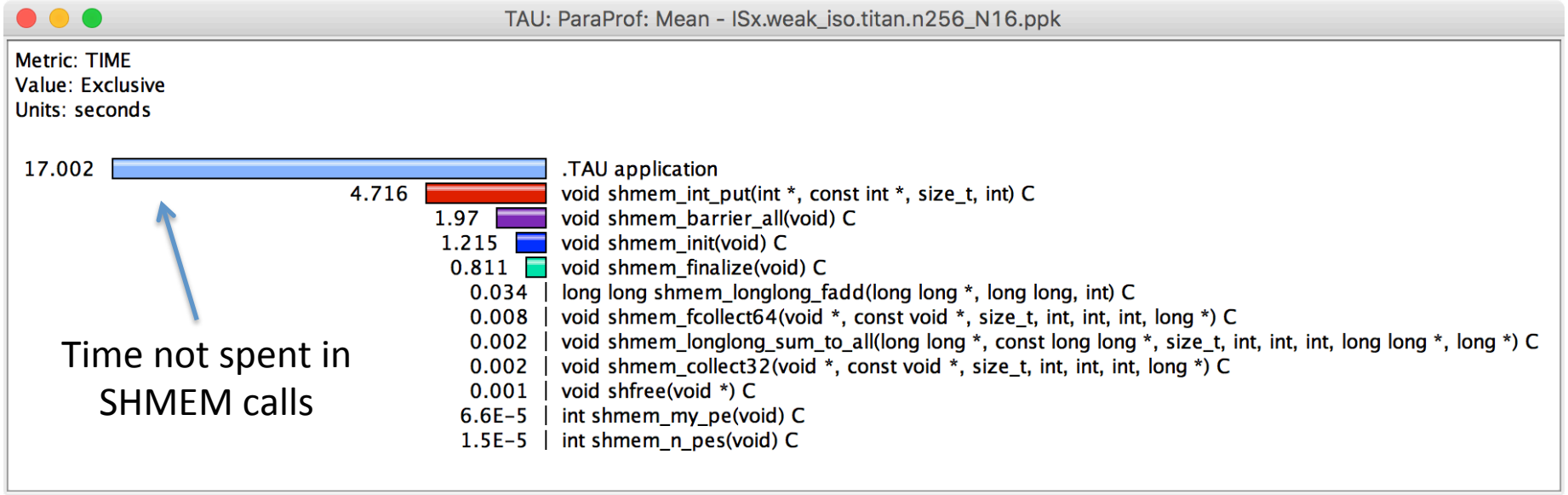
ISx Strong Scaling 256 Titan Cores



Copyright © ParaTools, Inc.

Time in SHMEM vs. Application

- ISx weak scaling isobuckets
- 256 PEs on 16 Titan nodes
- **`aprun -n 256 -N 16 tau_exec -shmem isx.weak_iso`**

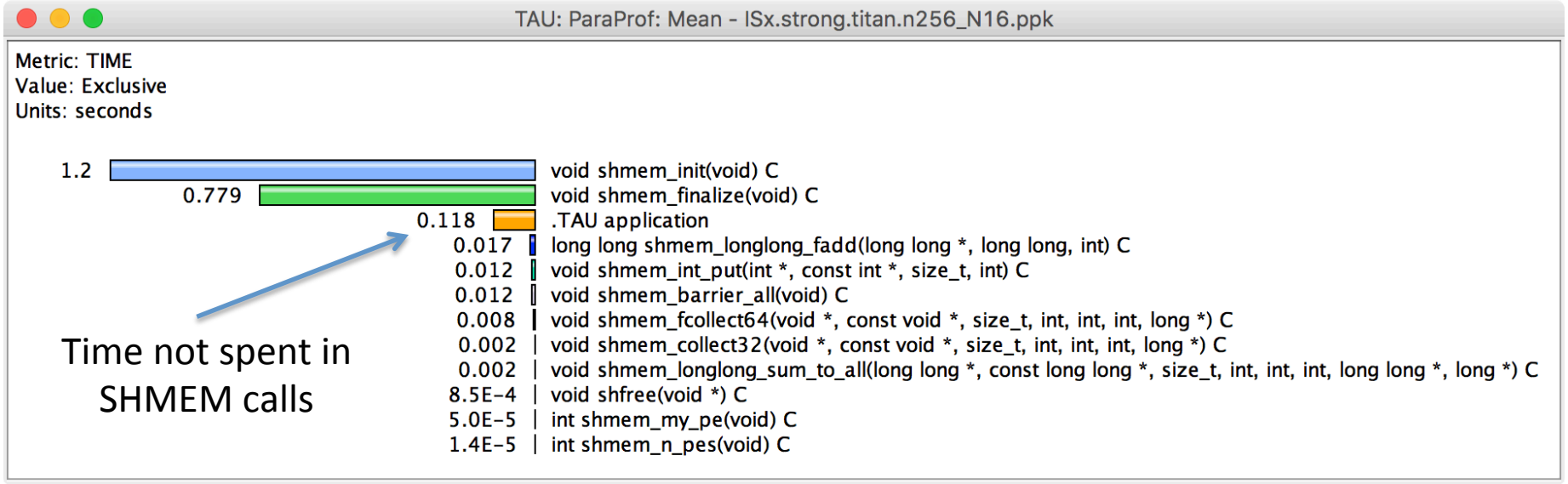


`% paraprof` **(Click on label, e.g. "Mean" or "node 0")**

Copyright © ParaTools, Inc.

Time in SHMEM vs. Application

- ISx strong scaling
- 256 PEs on 16 Titan nodes
- **`aprun -n 256 -N 16 tau_exec -shmem isx.strong`**



`% paraprof` **(Click on label, e.g. "Mean" or "node 0")**

Copyright © ParaTools, Inc.

Memory Allocation/Deallocation

TAU: ParaProf: Context Events for: node 0 - ISx.strong.titan.n256_N16.pdt.ppk

Name	Total	NumSamples	MaxValue	MeanValue	Std. Dev.
.TAU application					
Heap Memory Used (KB) at Entry	71.078	1	71.078	71.078	0
Heap Memory Used (KB) at Exit	2,095,745.812	1	2,095,745.812	2,095,745.812	0
Increase in Heap Memory (KB)	2,095,674.734	1	2,095,674.734	2,095,674.734	0.022
int main(const int, char **) C [{isx.c} {77,1}-{91,1}]					
Heap Memory Used (KB) at Entry	74.344	1	74.344	74.344	0
Heap Memory Used (KB) at Exit	2,095,743.594	1	2,095,743.594	2,095,743.594	0
Increase in Heap Memory (KB)	2,095,669.25	1	2,095,669.25	2,095,669.25	0.022
void log_times(char *) C [{isx.c} {583,1}-{615,1}]					
void init_shmem_sync_array(long *const restrict) C [{isx.c} {817,1					
int bucket_sort(void) C [{isx.c} {183,1}-{244,1}]					
char *parse_params(const int, char **) C [{isx.c} {96,1}-{174,1}]					
void shmem_init(void) C					
Heap Memory Used (KB) at Entry	78.5	1	78.5	78.5	0
Heap Memory Used (KB) at Exit	2,095,579.812	1	2,095,579.812	2,095,579.812	0
Increase in Heap Memory (KB)	2,095,501.312	1	2,095,501.312	2,095,501.312	0.022
void shmem_finalize(void) C					
Decrease in Heap Memory (KB)	8,386,709.938	3	4,193,279.984	2,795,569.979	1,976,659.551
Heap Memory Used (KB) at Entry	417,325,480,22...	201,120	2,096,798.172	2,075,007.36	300,712.116
Heap Memory Used (KB) at Exit	417,323,391,14...	201,120	2,096,798.172	2,074,996.973	300,892.442
Increase in Heap Memory (KB)	6,297,631.469	51	2,095,674.734	123,482.97	493,033.697
Message size received from all nodes	4,096	512	8	8	0
Message size sent to all nodes	2,093,136	1,022	4,416	2,048.078	2,046.067

% **`paraprof`** **(Right-click label [e.g "node 0"] → Show Context Event Window)**

Copyright © ParaTools, Inc.

Message Statistics

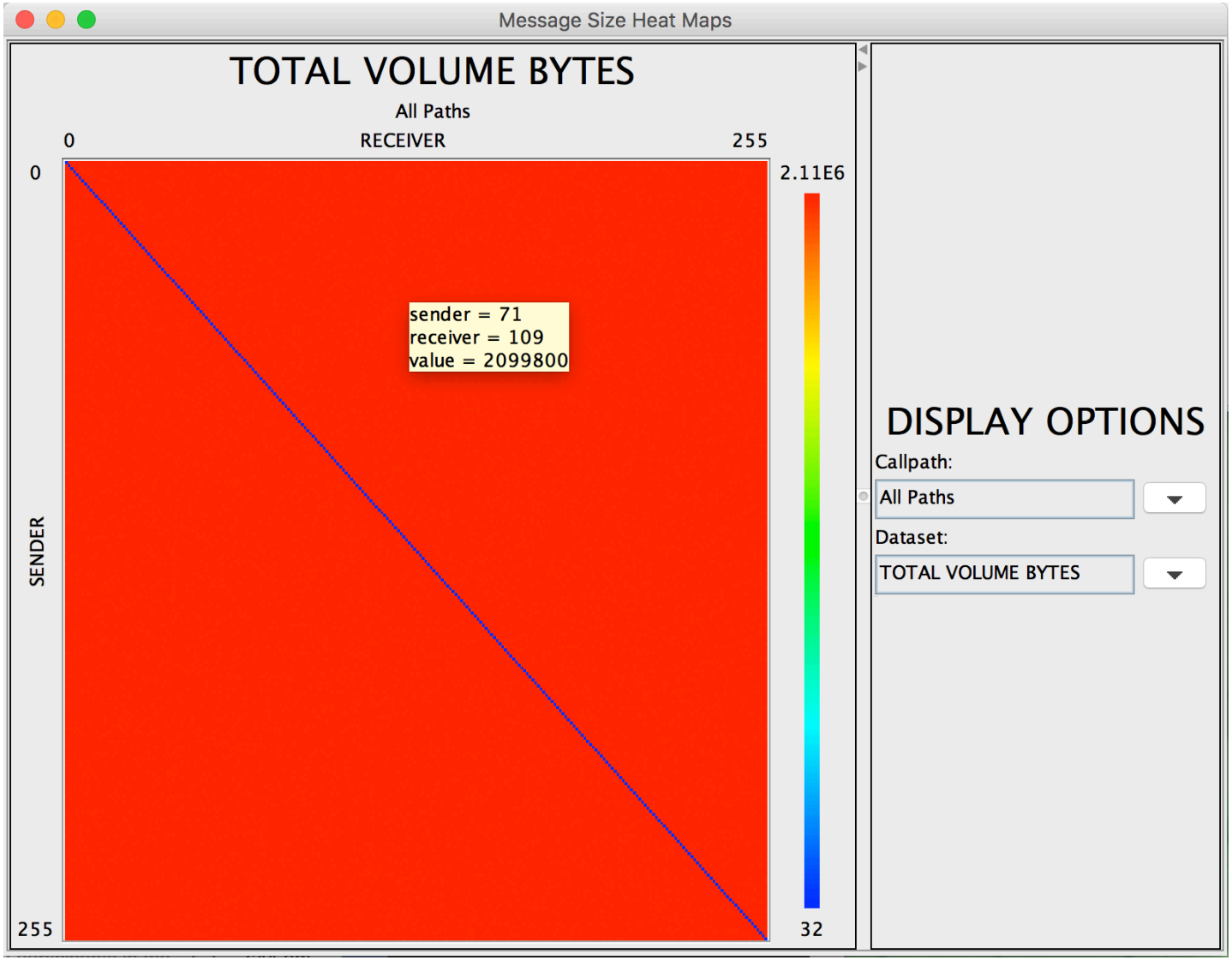
TAU: ParaProf: Context Events for: node 0 - ISx.strong.titan.n256_N16.ppk

Name	Total	NumSam...	MaxValue	MinValue	MeanValue	Std. Dev.
.TAU application						
Message size received from all nodes	4,096	512	8	8	8	0
Message size received from node 0	16	2	8	8	8	0
Message size received from node 1	16	2	8	8	8	0
Message size received from node 10	16	2	8	8	8	0
Message size received from node 100	16	2	8	8	8	0
Message size received from node 101	16	2	8	8	8	0
Message size received from node 102	16	2	8	8	8	0
Message size received from node 103	16	2	8	8	8	0
Message size received from node 104	16	2	8	8	8	0
Message size received from node 105	16	2	8	8	8	0
Message size received from node 106	16	2	8	8	8	0
Message size received from node 107	16	2	8	8	8	0
Message size received from node 108	16	2	8	8	8	0
Message size received from node 109	16	2	8	8	8	0
Message size received from node 11	16	2	8	8	8	0
Message size received from node 110	16	2	8	8	8	0
Message size received from node 111	16	2	8	8	8	0
Message size received from node 112	16	2	8	8	8	0
Message size received from node 113	16	2	8	8	8	0
Message size received from node 114	16	2	8	8	8	0
Message size received from node 115	16	2	8	8	8	0
Message size received from node 116	16	2	8	8	8	0
Message size received from node 117	16	2	8	8	8	0
Message size received from node 118	16	2	8	8	8	0
Message size received from node 119	16	2	8	8	8	0
Message size received from node 12	16	2	8	8	8	0
Message size received from node 120	16	2	8	8	8	0
Message size received from node 121	16	2	8	8	8	0
Message size received from node 122	16	2	8	8	8	0
Message size received from node 123	16	2	8	8	8	0

% **`paraprof`** **(Right-click label [e.g "node 0"] → Show Context Event Window)**

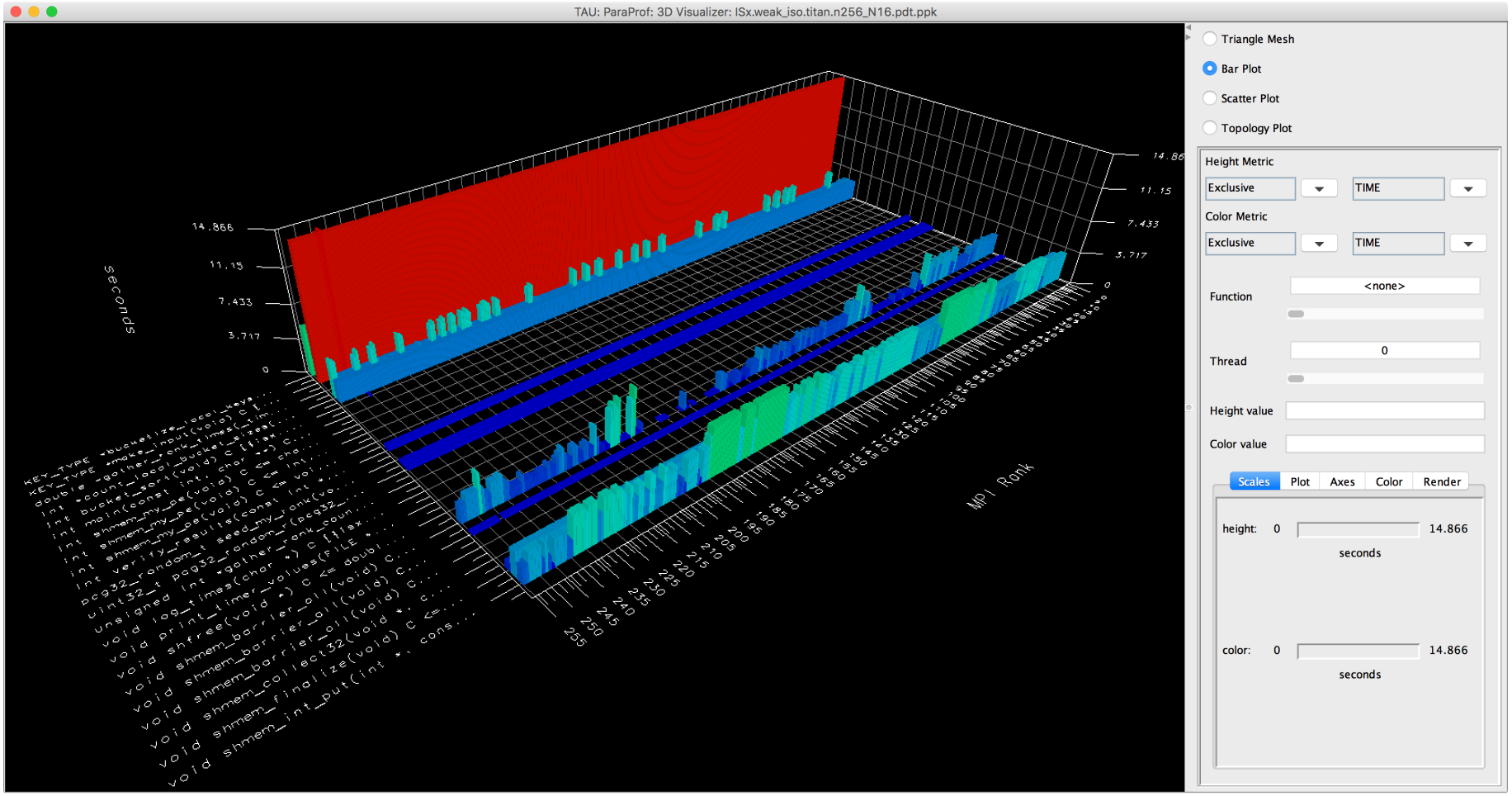
Copyright © ParaTools, Inc.

Communication Matrix



Copyright © ParaTools, Inc.

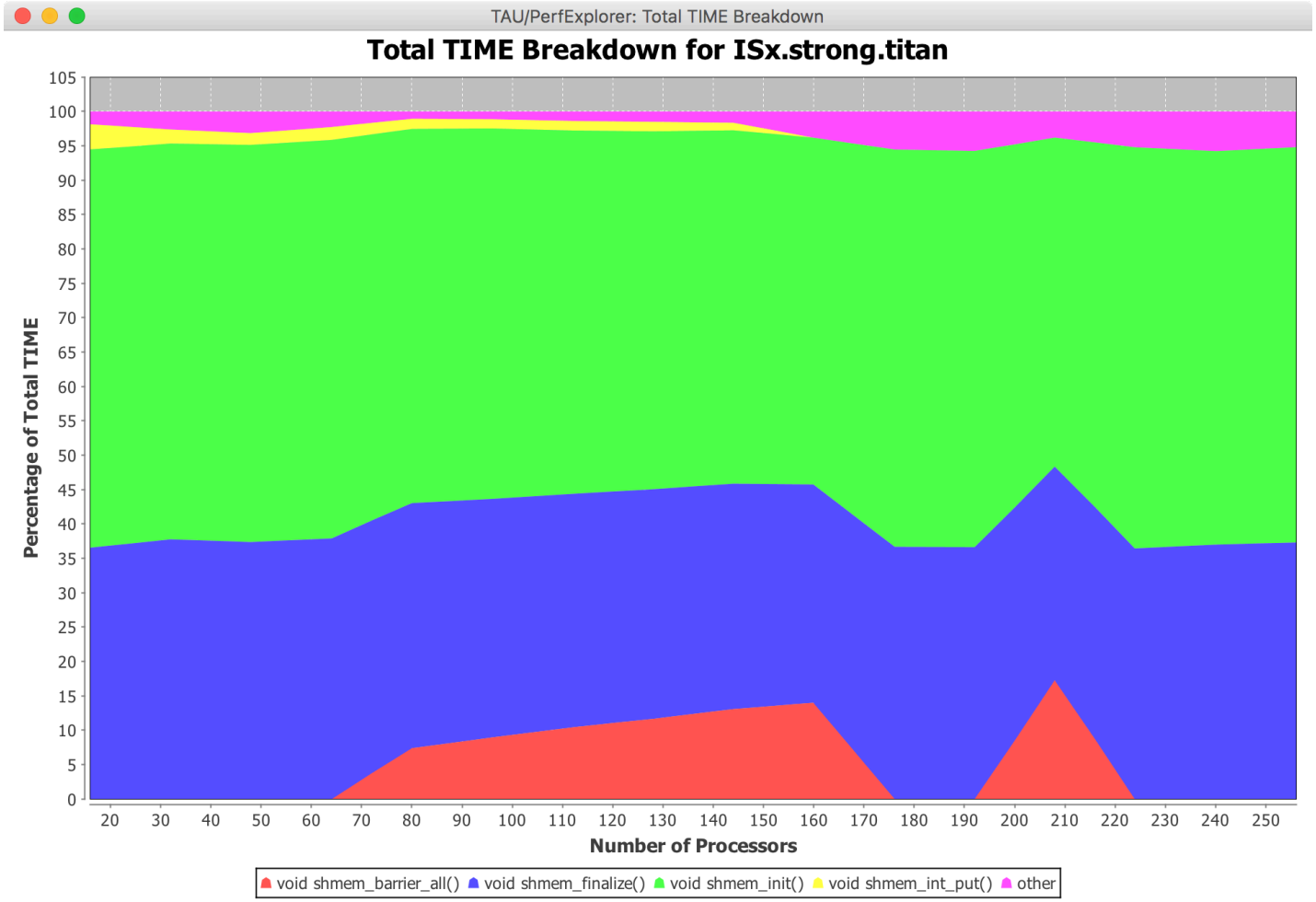
3D Profile Visualization



% paraprof (Windows → 3D Visualization)

Copyright © ParaTools, Inc.

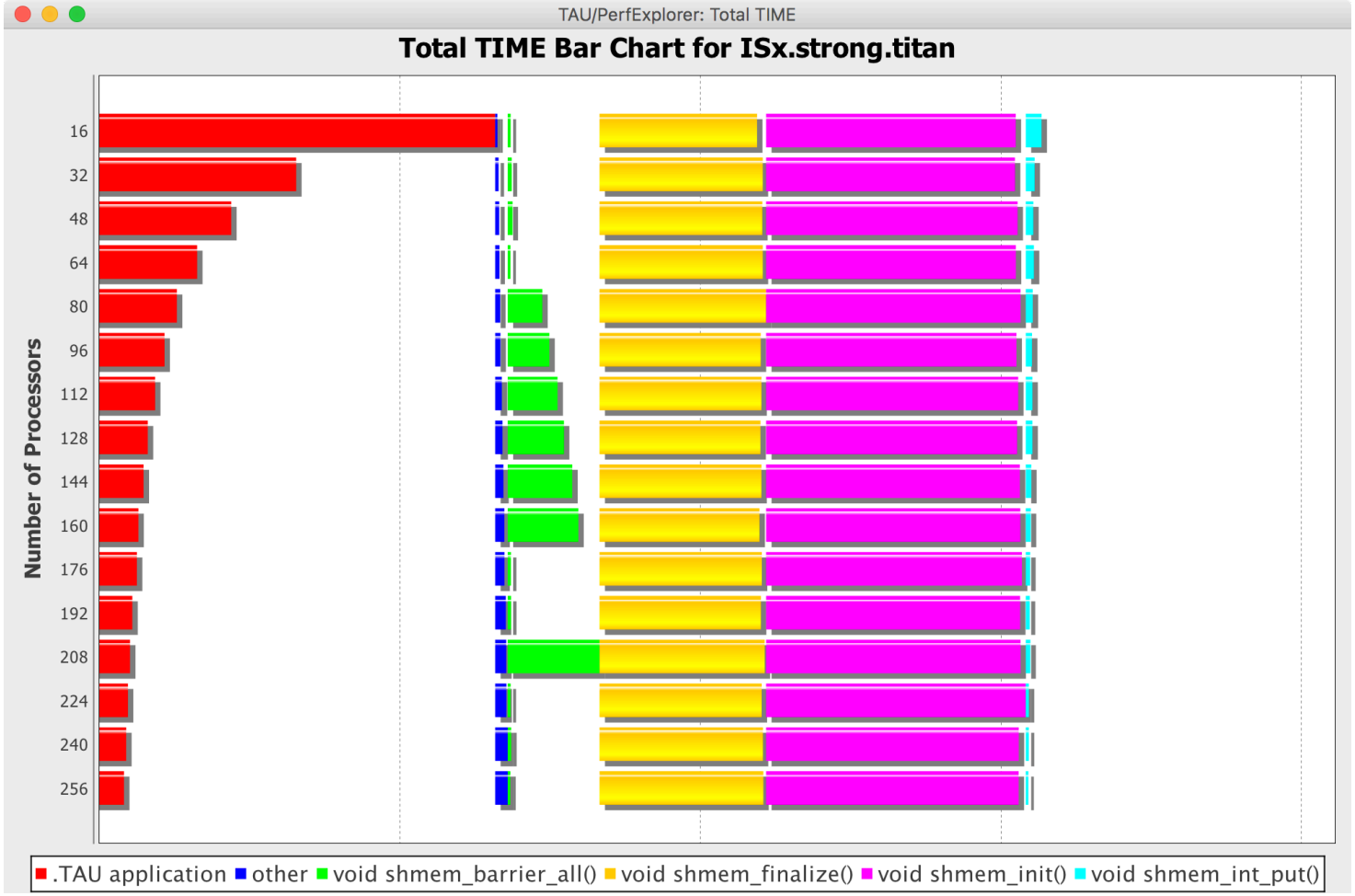
How Does Each Routine Scale?



`% perfexplorer` (Charts → Runtime Breakdown)

Copyright © ParaTools, Inc.

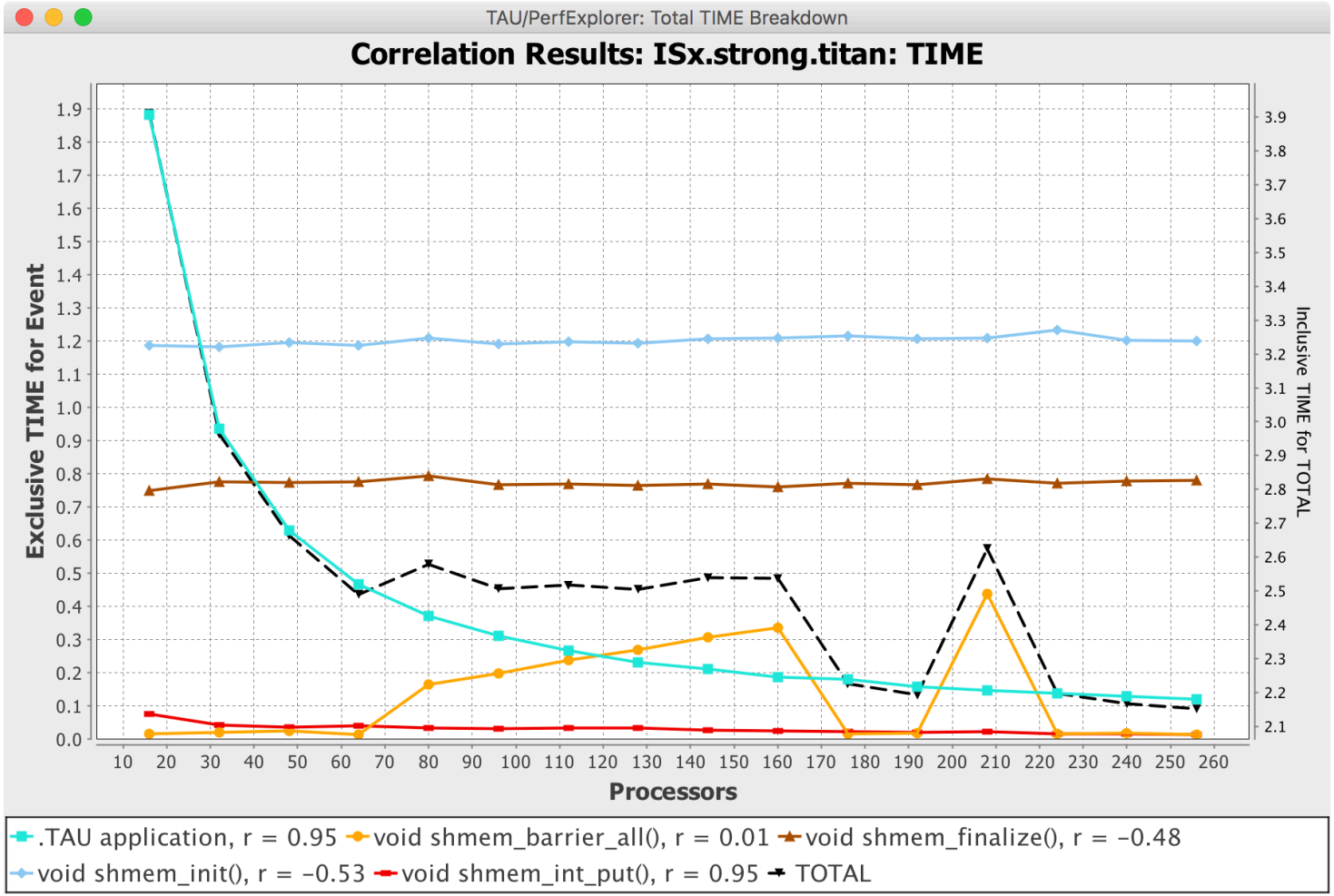
How Does Each Routine Scale?



`% perfexplorer` (Charts → Stacked Bar Chart)

Copyright © ParaTools, Inc.

Which Events Correlate with Runtime?



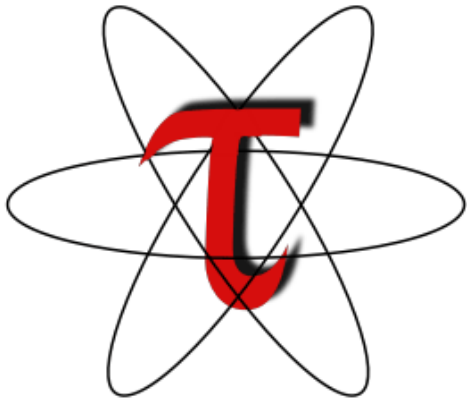
% `perfexplorer` (Charts → Correlate Events with Total Runtime)

Copyright © ParaTools, Inc.

CONCLUSION

Copyright © ParaTools, Inc.

Downloads



http://tau.uoregon.edu

http://github.com/ParaToolsInc/taucmdr

http://www.hpclinux.com

Free download, open source, BSD license

Copyright © ParaTools, Inc.